Image Inpainting Considering Symmetric Patterns

Norihiko Kawai and Naokazu Yokoya
Graduate School of Information Science, Nara Institute of Science and Technology
$\{$*norihi-k, yokoya*$\}$*@is.naist.jp*

Abstract

This paper proposes a novel image inpainting method to remove undesired objects in an image. Conventionally, missing regions are filled in using similar textures in an image as exemplars. However, unnatural textures are often generated due to the paucity of available samples. In this study, we take into account symmetric transformation of texture patterns to increase exemplars. To generate plausible textures in missing regions with variously transformed patterns, we employ two approaches: (1) we use spatial coherence of texture patterns when searching for similar patterns, and (2) we define a new degree of confidence of exemplars for determining pixel values. The effectiveness of the proposed method is demonstrated by comparing results by three methods.

1 Introduction

Image inpainting methods have been widely investigated to remove undesired objects in an image. These methods can be classified into two categories: One uses information around missing regions and the other uses exemplars in the rest of the image, which is referred to as data region. The former methods [2, 6] propagate pixel values from the boundary of the missing region to the inner part using partial differential equations and so on. These methods are effective for small gaps but cannot generate complex textures. On the other hand, the latter methods [1, 3, 4, 7, 8, 9, 10, 11] synthesize textures for missing regions based on pattern similarity between missing and data regions. The methods can generate complex and good textures for many images. However, unnatural textures are still generated due to the paucity of available samples in the image.

To increase available samples, there have already been some attempts in terms of optical and geometric expansion of patterns. As for optical expansion of texture patterns, our previous method [8] allows brightness transformation of texture patterns to utilize patterns with the same geometry but different brightness. As for geometric expansion, some methods [7, 10] use symmetric patterns in an image. These methods detect symmetric patterns using contours of objects before inpainting. Therefore, they are effective for an image including clear object contours with a relatively simple background. However, it is difficult to detect symmetry in natural images including various texture patterns.

In this paper, we propose a new image inpainting method which is based on using symmetric patterns without explicitly detecting symmetry by extending our previous method [8]. Considering the combination of variously transformed exemplars sometimes causes the production of unnatural textures, we propose two approaches to prevent unnatural textures. One is based on spatial coherence of PatchMatch [5]. The proposed method propagates not only pixel positions of similar patterns but also geometric transformation parameters in searching for good exemplars. The other is based on the idea that the degree of confidence of exemplars when determining pixel values is newly defined with three elements which are different from conventional one. These approaches result in generating natural textures at low calculation cost.

2 Proposed methodology

In the proposed method, after initial values are given to missing regions, the target regions are inpainted by minimizing an energy function. In the following sections, we describe the definition of an energy function and the minimization of the energy function.

2.1 Energy function considering symmetric patterns

As illustrated in Figure 1, an image is divided into region Ω' including missing region Ω and data region Φ, which is the rest of the image. An energy function is defined as follows:

$$E = \sum_{\mathbf{x}_i \in \Omega'} w_{\mathbf{x}_i} \left\{ SSD(\mathbf{x}_i, \mathbf{x}_j, \mathbf{T}_{\mathbf{x}_i \mathbf{x}_j}) + \kappa SD(\mathbf{x}_i, \mathbf{x}_j) \right\}, \quad (1)$$

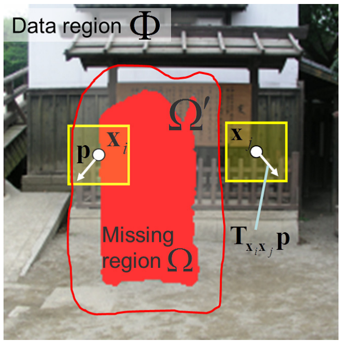


Figure 1. Data and missing regions.

where SSD is the pattern similarity considering brightness and symmetric transformation between missing and data regions. SD is the cost term considering spatial locality, which impose a constraint that similar texture patterns tend to exist nearer in an image. $\mathbf{x}_i$ is a pixel in region Ω' and $\mathbf{x}_j$ is a pixel in data region Φ. $w_{\mathbf{x}_i}$ is the weight for each pixel in region Ω'. Weight κ is the relative importance of SSD and SD.

The similarity measure SSD is defined as:

$$SSD(\mathbf{x}_i, \mathbf{x}_j, \mathbf{T}_{\mathbf{x}_i\mathbf{x}_j}) = \sum_{\mathbf{p}\in W} \left\{ I(\mathbf{x}_i + \mathbf{p}) - \alpha_{\mathbf{x}_i\mathbf{x}_j} I(\mathbf{x}_j + \mathbf{T}_{\mathbf{x}_i\mathbf{x}_j}\mathbf{p}) \right\}^2, \quad (2)$$

where $I(\mathbf{x}_i)$ represents the intensity of pixel $\mathbf{x}_i$. $\mathbf{p}$ is a shift vector in window W. $\alpha_{\mathbf{x}_i\mathbf{x}_j}$ is the brightness modification coefficient defined by the ratio of average pixel values around pixels $\mathbf{x}_i$ and $\mathbf{x}_j$ [8]. This coefficient adjusts the brightness of the texture in the data region to that in the missing region so as to prevent unnatural brightness changes in the inpainted image. $\mathbf{T}_{\mathbf{x}_i\mathbf{x}_j}$ is a symmetric transformation matrix.

Cost term SD for spatial locality of a texture pattern is defined based on the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ with the same definition as [8] when symmetric transformation parameter $\mathbf{T}_{\mathbf{x}_i\mathbf{x}_j}$ is a unit matrix. SD is a relatively high constant value when $\mathbf{T}_{\mathbf{x}_i\mathbf{x}_j}$ is not a unit matrix. By adding the constraint of spatial locality, untransformed textures that exist near missing regions are preferentially selected as exemplars.

2.2 Energy minimization

The energy function is minimized by iterating two processes: (i) searching for similar patterns and (ii) parallel update of all the pixel values using the similar patterns. These processes are described in the following.

2.2.1 Searching for similar patterns

In process (i), the position of the most similar pattern $\mathbf{x}_j = f(\mathbf{x}_i)$ corresponding to pixel $\mathbf{x}_i$ in missing regions and the geometric transformation parameter $\mathbf{T}_{\mathbf{x}_i\mathbf{x}_j} = g(\mathbf{T}_{\mathbf{x}_i f(\mathbf{x}_i)})$ should be basically determined so as to minimize the following expression.

$$SSD(\mathbf{x}_i, \mathbf{x}_j, \mathbf{T}_{\mathbf{x}_i\mathbf{x}_j}) + \kappa SD(\mathbf{x}_i, \mathbf{x}_j). \quad (3)$$

However, it takes much cost to exhaustively search for the optimal position and transformation parameter. Therefore, we determine the parameters with an approximate nearest-neighbor algorithm based on PatchMatch [5].

In this study, two scans, (a) from left to right, top to bottom, and (b) from right to left, bottom to top, are alternately iterated. Concretely, the following three steps determine parameters in case of scan order (a). The similar manner is employed in case of scan order (b)

(Step 1) Propagation of similar position and symmetric transformation parameter

This step propagates not only similar positions but also symmetric transformation parameters to obtain good exemplars. Position $f(\mathbf{x}_i)$ and transformation parameter $\mathbf{T}_{\mathbf{x}_i\mathbf{x}_j}$ are updated by selecting the values which minimize Eq. (3) from the following three options:

1. $f(\mathbf{x}_i)$ and $g(\mathbf{T}_{\mathbf{x}_i f(\mathbf{x}_i)})$ determined in the previous iteration.
2. $f(\mathbf{x}_i - \mathbf{u}) + g(\mathbf{T}_{(\mathbf{x}_i-\mathbf{u}) f(\mathbf{x}_i-\mathbf{u})})\mathbf{u}$ and $g(\mathbf{T}_{(\mathbf{x}_i-\mathbf{u}) f(\mathbf{x}_i-\mathbf{u})})$ where $\mathbf{u} = (1, 0)^\top$.
3. $f(\mathbf{x}_i - \mathbf{v}) + g(\mathbf{T}_{(\mathbf{x}_i-\mathbf{v}) f(\mathbf{x}_i-\mathbf{v})})\mathbf{v}$ and $g(\mathbf{T}_{(\mathbf{x}_i-\mathbf{v}) f(\mathbf{x}_i-\mathbf{v})})$ where $\mathbf{v} = (0, 1)^\top$.

(Step 2) Random search with the transformation parameter fixed

Step 2 attempts to improve position $f(\mathbf{x}_i)$ by testing a sequence of candidate offsets at an exponentially decreasing distance from $f(\mathbf{x}_i)$ with the same approach as [5] using transformation parameter $g(\mathbf{T}_{\mathbf{x}_i f(\mathbf{x}_i)})$ determined by Step 1.

(Step 3) Random search with various transformation parameters

Step 3 attempts to improve position $f(\mathbf{x}_i)$ and transformation parameter $\mathbf{T}_{\mathbf{x}_i\mathbf{x}_j}$ by testing multiple random positions and transformation parameters.

2.2.2 Updating of pixel values

In process (ii), all the pixel values $I(\mathbf{x}_i)$ are updated in parallel so as to minimize the energy function. Energy function E can be decomposed into element energy $E(\mathbf{x}_i)$ for each pixel in missing region Ω. Element energy $E(\mathbf{x}_i)$ is expressed as follows:

$$E(\mathbf{x}_i) = \sum_{\mathbf{p}\in W} w_{\mathbf{t}} \{ I(\mathbf{x}_i) - \alpha_{\mathbf{t} f(\mathbf{t})} I(f(\mathbf{t}) - g(\mathbf{T}_{\mathbf{t} f(\mathbf{t})})\mathbf{p}) \}^2,$$
$$\mathbf{t} = \mathbf{x}_i + \mathbf{p}. \quad (4)$$

Each element energy is independent in terms of pixel values in missing region Ω. Therefore, energy function E is minimized by independently minimizing each element energy $E(\mathbf{x}_i)$. Each pixel value $I(\mathbf{x}_i)$ in the missing region can be calculated in parallel as follows:

$$I(\mathbf{x}_i) = \frac{\sum_{\mathbf{p}\in W} w_{\mathbf{t}} \alpha_{\mathbf{t}f(\mathbf{t})} I(f(\mathbf{t}) - g(\mathbf{T}_{\mathbf{t}f(\mathbf{t})})\mathbf{p})}{\sum_{\mathbf{p}\in W} w_{\mathbf{t}}}, \quad \mathbf{t} = \mathbf{x}_i + \mathbf{p}. \tag{5}$$

Here, in conventional method [8], weight w is defined by the distance from the boundary of missing region Ω. However, this weight tends to cause the production of unnatural textures at the center of the missing region. To improve generated textures, we dynamically change the weight for updating pixel values every iteration so as to propagate plausible textures. We define weight $w_{\mathbf{x}_i}$ with three elements as follows:

$$w_{\mathbf{x}_i} = C(\mathbf{x}_i) S(\mathbf{x}_i)^{\lambda} K(\mathbf{x}_i)^{\mu}, \tag{6}$$

where λ and μ are weights for the relative importance of the three terms. $C(\mathbf{x}_i)$ is defined as c^n, where n is the number of pixels which are not in missing region Ω in window W and c is a constant. That means that the more pixels which is not in missing region Ω a window includes, the more confident the texture corresponding to the window is. $S(\mathbf{x}_i)$ is the similarity between textures around $\mathbf{x}_i$ and $f(\mathbf{x}_i)$, and defined as the reciprocal of $SSD(\mathbf{x}_i, f(\mathbf{x}_i), g(\mathbf{T}_{\mathbf{x}_i f(\mathbf{x}_i)}))$. $K(\mathbf{x}_i)$ is the texture complexity around $\mathbf{x}_i$, and defined as the sum of magnitude values calculated with a Sobel filter in window W. $S(\mathbf{x}_i)$ plays the role of making plausible textures propagate into missing regions. However, blurry textures are sometimes generated in the missing regions because SSD is very sensitive to high-frequency texture and tends to give smaller values to blurry textures. To make up for this problem, texture complexity $K(\mathbf{x}_i)$ prevents textures from being blurry in missing region Ω. In addition, $K(\mathbf{x}_i)$ encourages clear edges and complex textures, which are very noticeable to human eyes, to propagate into missing region Ω.

3 Experiments

To demonstrate the effectiveness of the proposed method, we have applied our present method to several images with 200×200 pixels. We used a PC with Core i7 3.47GHz CPU and 12GB memory.

Figure 2 shows the comparison of the results by three methods. Method A is our previous method [8] that does not consider symmetric transformation. Method B considers symmetric transformation but the weight when updating pixel values is based on the distance from the boundary of a missing region. Method C is the proposed method. As for the top image including symmetric patterns in Figure 2, Method A did not generate plausible textures at the bottom-left corner of the board because this method cannot use symmetric patterns as exemplar. The result by Method B also did not generate plausible textures though this method considers symmetric patterns. On the other hand, the proposed method generated plausible symmetric textures. As for the bottom image, Methods A and B did not generate plausible textures because of inappropriate weight for updating pixel values. On the other hand, the proposed method gave good results.

Figure 3 shows inpainted results for various images by the proposed method. As for top two images which include symmetric patterns, plausible textures were generated in the missing regions. As for the bottom-left image, a part of the tower was not produced. In this image, tree colors in the background are different between left and right sides. Therefore, texture in the right side of the tower did not correspond to the left side with similarity measure SSD. To obtain good results for such scenes, we should consider a different similarity measure or segmentation of foreground and background. As for the bottom-right image which does not include symmetric patterns, the proposed method obtained as good results as previous method [8] without adverse effects. As for computational time, it took 4.6, 4.3, 9.1 and 9.5 seconds to inpaint missing regions in top-left, top-right, bottom-left and bottom-right images in Figure 3, respectively.

4 Conclusion

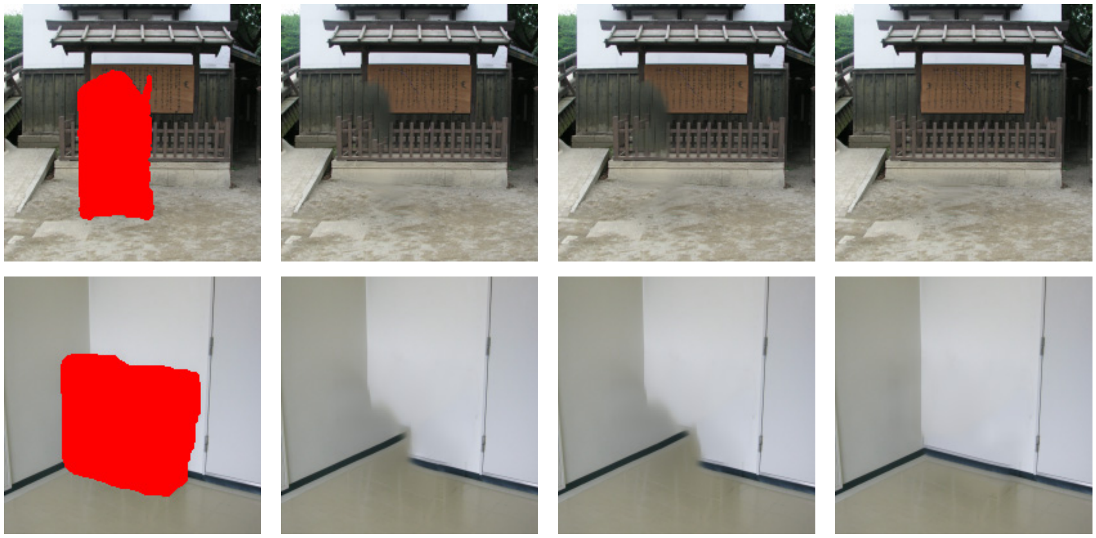
In this paper, we have proposed a new energy function considering symmetric patterns for image inpainting. To obtain good results for many images, two factors were considered: (1) propagation of similar texture positions and symmetric transformation parameters, and (2) weight for updating pixel values. In experiments, we demonstrated the effectiveness of the proposed method using several images. In future work, we should consider a similarity measure applicable to various scenes. In addition, we will consider additional geometric transformation.

Acknowledgement This work was partially supported by JSPS KAKENHI Nos. 23240024 and 24700118.

References

[1] A. Criminisi, P. Pérez, and K. Toyama. Region Filling and Object Removal by Exemplar-Based Image Inpainting. In *Trans. on Image Processing*, volume 13, No. 9, pages 1200–1212, 2004.

[2] A. Levin, A. Zomet, and Y. Weiss. Learning How to Inpaint from Global Image Statistics. In *Proc. ICCV*, volume 1, pages 305–312, 2003.



(a) Input with a missing region (red) (b) Method A [8] (c) Method B (d) Method C

Figure 2. Comparison of results inpainted by different methods.

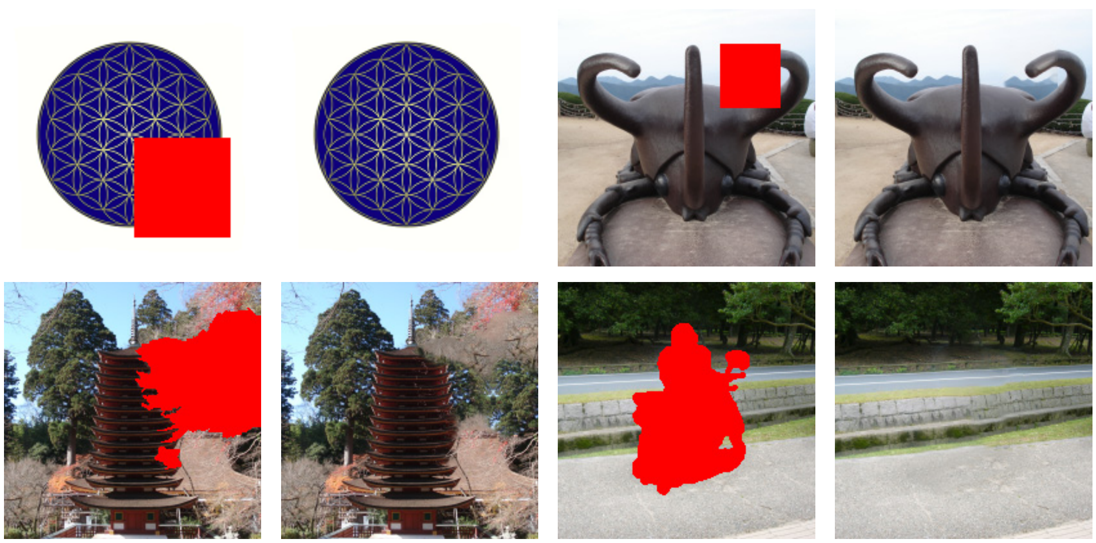


Figure 3. Inpainted results for various images by the proposed method.

[3] A.A. Efros and T.K. Leung. Texture Synthesis by Non-parametric Sampling. In *Proc. ICCV*, pages 1033–1038, 1999.

[4] C. Allène and N. Paragios. Image Renaissance Using Discrete Optimization. In *Proc. ICPR*, pages 631–634, 2006.

[5] C. Barnes, E. Shechtman, and A. Finkelstein. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. In *Proc. SIGGRAPH*, volume 28, 2009.

[6] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image Inpainting. In *Proc. SIGGRAPH*, pages 417–424, 2000.

[7] M.V. Venkatesh and S.S. Cheung. Symmetric shape completion under severe occlusions. In *Proc. ICIP*, pages 709–712, 2006.

[8] N. Kawai, T. Sato, and N. Yokoya. Image Inpainting Considering Brightness Change and Spatial Locality of Textures and Its Evaluation. In *Proc. PSIVT*, pages 271–282, 2009.

[9] N. Komodakis and G. Tziritas. Image Completion Using Global Optimization. In *Proc. CVPR*, pages 442–452, 2006.

[10] T. Pereira, R.P Leme, L. Velho, and T. Lewiner. Symmetry-based Completion. In *Proc. GRAPP*, pages 39–45, 2009.

[11] Y. Wexler, E. Shechtman, and M. Irani. Space-Time Completion of Video. In *Trans. on PAMI*, volume 29, No. 3, pages 463–476, 2007.